

Computing the state difference equations for discrete overdetermined linear mD systems

Kim Batselier^a, Ngai Wong^a,

^a*The Department of Electrical and Electronic Engineering, The University of Hong Kong*

Abstract

We derive an algorithm that computes the state difference equations for a given set of poles of linear discrete overdetermined autonomous mD systems. These difference equations allow the realization of the dynamical system by means of delay, multiplication and addition elements in simulation diagrams. In doing so we generalize the classical Cayley-Hamilton theorem to multivariate polynomial ideals and provide a system theoretic interpretation to the notion of polynomial ideals, leading monomials and Gröbner bases. Furthermore, we extend the problem to include poles at infinity and so arrive at a description of overdetermined descriptor systems. This results in a new state space description of autonomous mD descriptor systems. In addition, we discuss the separation of the state variables of singular mD systems into a regular and singular part. A sufficient condition under which these two state vector parts can be interpreted as a forward evolving regular part and a backward evolving singular part is given. The robustness and efficiency of the developed algorithms are demonstrated via numerical experiments.

Key words: time-invariant; n-dimensional systems; descriptor systems; computational methods; Gröbner bases.

1 Introduction

There has been an increasing interest in multi-dimensional (mD) systems in the systems and control community over the past decades [2,5,6,11,15,12,26]. Different formulations of mD systems were given [2,12,18,21,26] and concepts such as realization, stability, reachability and observability have been thoroughly analysed. In the 1D case, the poles of the system determine the state recursion equation. This fact is used for the design of linear filters or for the stabilization of the linear system by state feedback (the pole placement problem). In this article we extend the theory on how to determine the state difference equations for a given set of poles to the following class of discrete overdetermined multiple input-multiple output (MIMO) mD

linear systems

$$\begin{aligned} x(t_1+1, \dots, t_m) &= A_1 x(\mathbf{t}) + B_1 u(\mathbf{t}), \\ &\vdots \\ x(t_1, \dots, t_m+1) &= A_m x(\mathbf{t}) + B_m u(\mathbf{t}), \\ y(\mathbf{t}) &= C x(\mathbf{t}) + D u(\mathbf{t}), \end{aligned} \quad (1)$$

where we introduce the shorthand notation $\mathbf{t} \triangleq (t_1, \dots, t_m)$ and $x(\mathbf{t}) \in \mathbb{R}^n$, $u(\mathbf{t}) \in \mathbb{R}^p$ and $y(\mathbf{t}) \in \mathbb{R}^l$. The matrices $A_1, \dots, A_m, B_1, \dots, B_m, C, D$ have the appropriate dimensions. Note how the updating of the whole state $x(\mathbf{t})$ is described by a separate state equation for each independent direction t_1, \dots, t_m . It is for this reason that (1) are called overdetermined mD systems. Continuous overdetermined systems are characterized by having partial derivatives of the state with respect to t_1, \dots, t_m on the left hand side. These systems were originally studied in the field of operator theory [22,27]. More recently, the pole placement problem for state feedback of overdetermined continuous 2D systems was fully solved [28]. Discrete overdetermined systems as described by (1) have only been studied in recent years [5,6,11,15]. They have found direct applications in model order reduction [5] and polynomial optimization [6]. Furthermore, the relations between autonomous overdetermined mD systems, solving multivariate poly-

* This work was supported in part by the Hong Kong Research Grants Council under General Research Fund (GRF) Projects 17208514 and 17212315, and the University Research Committee of The University of Hong Kong.

Email addresses: kim.batselier@eee.hku.hk (Kim Batselier), nwong@eee.hku.hk (Ngai Wong).

nomial systems, Gröbner bases and systems theory are discussed in [11,15].

Other commonly used m D state space models are the Roesser model [26] and the Fornasini-Marchesini model [12]. It is important to observe that the state space system (1) is quite different from the Roesser model in that the state vector $x(t)$ is not divided into different partial state vectors along each dimension. It is also not required to have an infinite amount of initial states in order to compute the state recursion. In fact we will show that, just like in the 1D case, the system order n denotes the total number of previous states required in the state difference equations in order to compute the next state. It is possible for the 2D case to write (1) in terms of the general model of [21], which generalizes the Roesser, Fornasini-Marchesini and Attasi models. Indeed, the general model is

$$\begin{aligned} x(t_1+1, t_2+1) = & G_0 x(t_1, t_2) + G_1 x(t_1, t_2+1) \\ & + G_2 x(t_1+1, t_2) + H_0 u(t_1, t_2) \\ & + H_1 u(t_1, t_2+1) + H_2 u(t_1+1, t_2), \end{aligned}$$

which is equivalent with (1) if $G_0 = A_1 A_2 = A_2 A_1$, $H_0 = (A_1 B_2 + A_2 B_1)/2$, $G_1 = G_2 = 0$ and $H_1 = B_1/2$, $H_2 = B_2/2$ are satisfied. However, the general model of [21] still needs an infinite amount of initial states in order to compute all future states. Note that the A_i matrices also commute for the Attasi models [2].

The main contribution of this article is the derivation and implementation of a numerical algorithm that solves the following problem:

Problem 1 *Given the finite set of poles \mathcal{Z} of (1), find a minimal set of difference equations $G = \{g_1, \dots, g_k\}$ that allow us to recursively compute all future states of the corresponding autonomous m D system through monomial shifts $z_1^{t_1} \dots z_m^{t_m}$ of g_1, \dots, g_k .*

Consequently, these future states completely determine the free response ($u(t) \equiv 0 \forall t_1, \dots, t_m \geq 0$) of the overdetermined m D system (1). Problem 1 is trivially solved for the 1D case. Indeed, the unique monic characteristic polynomial $g_1(z_1)$ of A_1 in the shift operator z_1 is $(z_1 - \lambda_1) \dots (z_1 - \lambda_n) = z_1^n + c_{n-1}z_1^{n-1} + \dots + c_0$. The Cayley-Hamilton theorem allows us to write

$$g_1(A_1) = A_1^n + c_{n-1}A_1^{n-1} + \dots + c_0I_n = 0,$$

which can be post-multiplied with $x(0)$ to obtain

$$g_1 = x(n) + c_{n-1}x(n-1) + \dots + c_0x(0) = 0, \quad (2)$$

since $x(t) = A_1^t x(0)$ for an autonomous system. The linear relation (2) can be interpreted as a way to compute $x(n)$ from the known states $x(n-1), \dots, x(0)$ without needing to make an explicit choice of a basis. Consequently, all future states $x(t)$ ($t > n$) are found from multiplying (2) with powers of z_1 (or A_1). Indeed, $x(n+1)$

can be computed from multiplying (2) with z_1 , $x(n+2)$ from multiplying (2) with z_1^2 and so forth. In other words, the set of all polynomials in z_1 from which all future states $x(t)$ can be computed form a polynomial ideal $\langle p(z_1) \rangle$. Generalizing this result to the m D case ($m > 1$) is less trivial. We will show that for the m D case the characteristic polynomial of A_1 needs to be replaced by a Gröbner basis. Essentially, solving this problem is equivalent with computing a Gröbner basis that vanishes on the given set of poles \mathcal{Z} . The first symbolical algorithm that solves this problem was developed by Möller and Buchberger [25]. A numerical implementation of the affine Buchberger-Möller algorithm is described in [16]. Their numerical implementation uses both the singular value decomposition (SVD) and Gaussian Elimination (GE). We will derive our own version of both the affine and projective Buchberger-Möller algorithm, using only the SVD, from a system theoretic point of view.

Once a Gröbner basis g_1, \dots, g_k is found that vanishes on the given set of poles \mathcal{Z} , we then have a set of polynomials that generate the polynomial ideal $\langle g_1, \dots, g_k \rangle$ of state difference polynomials. The generalization of the Cayley-Hamilton theorem then states that the A_i matrices of the m D system (1) satisfy all polynomial equations $p(A_1, \dots, A_m) = 0$ with $p \in \langle g_1, \dots, g_k \rangle$. Many generalizations of the Cayley-Hamilton theorem have appeared [21,19,29], but none from the point of view of Gröbner bases and systems theory. More theory on Gröbner bases and their application in systems theory can be found in [7,8]. Poles at infinity of linear systems have been extensively studied and it is commonly known that poles at infinity are intimately linked with the notion of singular or descriptor systems [18,23,24]. These notions will also be generalized here to the m D case, leading to a new state space description of singular m D systems in terms of a homogenization variable t_0 .

The inverse problem of Problem 1 has received more attention in the literature. It is the realization problem of finding a set of A_i matrices from a given set of difference equations [13] or m D data trajectories [30], which have both been solved in the behavioural context. An alternative realization method that does not rely on the behavioural context, nor on the computation of a Gröbner basis, is a generalization of the Ho-Kalman realization algorithm [17] to the m D case [11]. This problem is trivially solved when the poles \mathcal{Z} are given. Indeed, one can then construct the A_i matrices in their diagonal or upper triangular form using the method described in [4]. The problem solved in this article is precisely the inverse of the aforementioned realization problem: from a given set of poles \mathcal{Z} , find the minimal set of state difference equations. These state difference equations can then be easily realized with delay, multiplication and addition elements into a simulation diagram.

This article is organized as follows. In Section 2 we establish the notation and define some required concepts from algebraic geometry in the context of systems theory. In Section 3 we define regular systems and derive the algorithm to compute all state difference polynomials from

a given set of poles. The conceptual changes necessary to extend the algorithm to work for singular systems are discussed in Section 4. The application of the algorithms for both regular as for singular systems are demonstrated in Section 5. All algorithms are implemented in MATLAB and Octave and are freely available at https://github.com/kbatseli/PNLA_MATLAB_OCTAVE.

2 Preliminaries

2.1 State orderings

All signals and state vectors are indexed by m independent variables t_1, \dots, t_m . Hence, in order to be able to define a state sequence, one needs to introduce a total ordering on the m -tuples $\mathbf{t} \in \mathbb{N}_0^m$. For notational convenience, we introduce the shorthand notation $x(\mathbf{0})$ for the initial state $x(0, \dots, 0)$. Then, by introducing the m linear shift operators z_1, \dots, z_m such that

$$z_k x(\mathbf{t}) = x(t_1, \dots, t_k + 1, \dots, t_m) \quad (1 \leq k \leq m),$$

we can write

$$x(\mathbf{t}) = z_1^{t_1} \dots z_m^{t_m} x(\mathbf{0}) = A_1^{t_1} \dots A_m^{t_m} x(\mathbf{0}), \quad (3)$$

for the autonomous version of (1). The A_i matrices are in fact nothing but the representations of the z_i operators with respect to a particular basis. Since the order in which the z_i operators are applied should not matter, it immediately follows that the A_i matrices commute. This will be further explained in Section 3. It is also straightforward to see that there is a bijection between \mathbf{t} and the monomials $z^{\mathbf{t}} \triangleq z_1^{t_1} \dots z_m^{t_m}$. Therefore, any ordering on the monomials $z_1^{t_1} \dots z_m^{t_m}$ immediately defines an ordering on the m -tuples \mathbf{t} and consequently determines the state sequence. There are many possible monomial orderings [10, p. 54]. We will use the graded xelicographic ordering [3, p. 104] throughout this article. This ordering orders monomials \mathbf{a}, \mathbf{b} first by means of their total degree $|\mathbf{a}|, |\mathbf{b}|$. If these total degrees are equal, then the xelicographic ordering $>_{xel}$ is used. For example, using this monomial ordering in \mathbb{N}_0^2 leads to the following state sequence

$$x(0, 0) < x(1, 0) < x(0, 1) < x(2, 0) < x(1, 1) < \dots$$

Choosing a particular monomial ordering also allows us to determine which monomial of a multivariate polynomial p is the leading monomial, denoted $\text{LM}(p)$.

2.2 Polynomial ideals and Gröbner bases

The poles of a 1D linear time-invariant system are defined as the roots of $\det(z_1 I - A_1)$. We first extend this definition to the m D case. Note that since the matrices A_1, \dots, A_m in (1) commute, this means that they have n common eigenvectors.

Definition 2.1 Suppose we have a set of commuting matrices A_1, \dots, A_m that describe an m D system (1). Then for each common eigenvector v of A_1, \dots, A_m we define the corresponding pole as $\lambda = (\lambda_1, \dots, \lambda_m) \in \mathbb{C}^m$, where each component λ_i satisfies $A_i v = \lambda_i v$ ($i = 1, \dots, m$). The set that contains the n poles of (1) is denoted \mathcal{Z} .

As in the 1D case, polynomials p_1, \dots, p_s in the shift operators z_1, \dots, z_m that vanish on the poles \mathcal{Z} will allow us to express a next state $\text{LM}(p_1), \dots, \text{LM}(p_s)$ as a linear combination of the previous states. Indeed, for the 1D case we can rewrite (2) as

$$\text{LM}(g_1) = x(n) = -c_{n-1}x(n-1) - \dots - c_0x(0).$$

Unlike the 1D case, multiplying these polynomials p_1, \dots, p_s with all m -variate monomials $z_1^{t_1} \dots z_m^{t_m}$ is in general not sufficient to describe all future states. Indeed, take for example $p_1 = 1 + 2z_1 - 3z_2$ and $p_2 = -2 + z_1 + z_2$. Then a straightforward computation shows that $(1 + z_1)p_1 + (3 + 3z_1)p_2 = -5 + 5z_1^2$ also vanishes on the roots of p_1, p_2 and has a leading monomial z_1^2 that is not a multiple of $\text{LM}(p_1)$ or $\text{LM}(p_2)$. This observation tells us that in order to describe all future states of an autonomous overdetermined m D system, one needs to consider the polynomial ideal generated by a given set of state difference polynomials.

Definition 2.2 ([10, p. 30]) Let $p_1, \dots, p_s \in \mathcal{C}^m$ be a polynomial system with solution set \mathcal{Z} , then

$$\mathcal{I} = \langle p_1, \dots, p_s \rangle \triangleq \left\{ f = \sum_{i=1}^s h_i p_i \mid h_1, \dots, h_s \in \mathcal{C}^m \right\}$$

is the polynomial ideal generated by p_1, \dots, p_s and \mathcal{C}^m denotes the ring of m -variate polynomials with complex coefficients. If only the solution set \mathcal{Z} is given, then

$$\mathcal{I}(\mathcal{Z}) \triangleq \left\{ f \in \mathcal{C}^m \mid f(z) = 0 \forall z \in \mathcal{Z} \right\}$$

is the corresponding polynomial ideal of all polynomials that vanish on \mathcal{Z} .

Now, given a polynomial ideal $\mathcal{I} = \langle p_1, \dots, p_s \rangle$, where each of the polynomials p_1, \dots, p_s is to be interpreted as a difference equation on the state x , then the set of all possible future states is the set of all leading terms in \mathcal{I} .

Definition 2.3 Let $p_1, \dots, p_s \in \mathcal{C}^m$ be a polynomial system with solution set \mathcal{Z} , then

$$\text{LM}(\mathcal{I}) \triangleq \{ \text{LM}(p) \mid \forall p \in \langle p_1, \dots, p_s \rangle \} \quad (4)$$

is the set of all future states that are determined by p_1, \dots, p_s .

The Hilbert Basis Theorem [10, p. 76] states that every polynomial ideal $\mathcal{I}(\mathcal{Z})$ determined by a finite set of roots (poles) \mathcal{Z} has a finite generating set p_1, \dots, p_s . A particular generating set with a special property is the Gröbner basis.

Definition 2.4 ([10, p. 77]) *Fix a monomial order. A finite subset $G = \{g_1, \dots, g_k\}$ of an ideal \mathcal{I} is a Gröbner basis if $\langle \text{LM}(g_1), \dots, \text{LM}(g_k) \rangle = \langle \text{LM}(\mathcal{I}) \rangle$.*

Hence, a Gröbner basis G has the useful property that all future states are monomial multiples of its leading monomials. It is important to note that the chosen monomial ordering determines the Gröbner basis. Furthermore, a Gröbner basis is called reduced if the coefficient of all its leading terms is 1 and no monomial in any element of the basis is divisible by the leading monomials of the other elements of the basis. These monomials b_i that are not divisible by any of the leading monomials $\text{LM}(g_1), \dots, \text{LM}(g_k)$ are also called the standard monomials [9, p. 36] and their set B is therefore described by

$$B = \{b_i = x^{\alpha_i} : b_i \notin \langle \text{LM}(\mathcal{I}) \rangle\},$$

which means that the standard monomials are the ‘initial’ states that are required to start up the state sequence using the state difference polynomials. Since the polynomial ideal $\mathcal{I}(\mathcal{Z})$ is radical, we also have that the total number of standard monomials equals the number of poles n [10, p. 235] (Proposition 8). If we denote the set of leading monomials of a Gröbner basis by $A = \{a_1, \dots, a_k\}$, then its corresponding reduced Gröbner basis are polynomials of the form

$$g_i = a_i + \sum_{j=1}^n \beta_j b_j \quad (\beta_j \in \mathbb{C}, i = 1, \dots, k). \quad (5)$$

Just as in the 1D case, the order n of the system is the total number of previous states required in the difference equations g_1, \dots, g_k to compute a next state. A reduced Gröbner basis has the useful property that it is unique. Once the initial states B and the difference equations g_1, \dots, g_k are known, one can realize the overdetermined autonomous m D systems in terms of delay, multiplication and addition elements. We illustrate the realization of a 2D system in Section 5. In Sections 3 and 4 we derive a numerical algorithm that computes a reduced Gröbner basis of state difference equations for a given set of poles \mathcal{Z} for both regular and a singular overdetermined m D systems.

2.3 Homogeneous polynomials and projective space

In order to describe poles at infinity and the resulting descriptor systems we need to consider homogeneous polynomials and projective coordinates. A polynomial of degree d is homogeneous when every term is of degree d .

The set of all homogeneous polynomials in $m + 1$ variables is denoted \mathcal{P}^m . A non-homogeneous polynomial $p \in \mathbb{C}^m$ of degree d can easily be made homogeneous by introducing an extra homogenization variable z_0 and multiplying each term of p with a power of z_0 such that its degree becomes d . Now consider the following equivalence relation \sim on the nonzero points of \mathbb{C}^{m+1} :

$$\begin{aligned} (z'_0, \dots, z'_m) &\sim (z_0, \dots, z_m) \\ \Leftrightarrow (z'_0, \dots, z'_m) &= \alpha (z_0, \dots, z_m) \end{aligned}$$

for some nonzero $\alpha \in \mathbb{C}$. The projective space is then defined as the set of resulting equivalence classes in the following way.

Definition 2.5 ([10, p. 368]) *The m -dimensional projective space \mathbb{P}^m is the set of equivalence classes of \sim on $\mathbb{C}^{m+1} - \{0\}$. Each nonzero $(m + 1)$ -tuple (z_0, \dots, z_m) defines a point z in \mathbb{P}^m , and we say that (z_0, \dots, z_m) are homogeneous coordinates of z .*

Note that the origin $\mathbf{0} \in \mathbb{C}^{m+1}$ is not a point in the projective space. Because of the equivalence relation \sim , an infinite number of projective points (z_0, \dots, z_m) can be associated with 1 affine point (z_1, \dots, z_m) . The affine space \mathbb{C}^m can be retrieved from the projective space by setting $z_0 = 1$. This means that given a projective point $z = (z_0, \dots, z_m)$ with $z_0 \neq 0$, its affine counterpart is $(1, z_1/z_0, \dots, z_m/z_0)$. The projective points for which $z_0 = 0$ are called points at infinity.

3 Regular m D systems

Before going into the derivation of the algorithm to compute a Gröbner basis from a given set of poles, we first define a regular overdetermined m D system.

Definition 3.1 *A regular overdetermined m D system is a system for which all poles \mathcal{Z} are points in the affine space \mathbb{C}^m .*

A regular m D system has therefore per definition no poles at infinity. One could be tempted to think that the m univariate characteristic polynomials $P = \{p_1(z_1), \dots, p_m(z_m)\}$ of their respective A_1, \dots, A_m matrices are the desired state difference polynomials. If this were the case, then the n poles of \mathcal{Z} should be the only roots of the polynomial system P . However, from Bézout’s Theorem [9, p. 115] it follows that P has n^m roots and therefore P is not the desired set of polynomials. In order to determine the desired state difference polynomials, one needs to consider the following state sequence matrix

$$X_d = \begin{pmatrix} x(0, \dots, 0) & x(1, \dots, 0) & \dots & x(0, \dots, d) \end{pmatrix}, \quad (6)$$

which contains the state sequence, according to the chosen monomial ordering, starting from the initial state

with $|\mathbf{t}| = 0$ up to all states with $|\mathbf{t}| = d$. The matrix X_d is easily constructed in practice from an initial state $x(\mathbf{0})$ using (3) as

$$X_d = \begin{pmatrix} x(\mathbf{0}) & A_1 x(\mathbf{0}) & A_2 x(\mathbf{0}) & \cdots & A_m^d x(\mathbf{0}) \end{pmatrix}. \quad (7)$$

Since the components of each of the poles λ are known, the matrices A_1, \dots, A_m are also easily constructed in diagonal form. When some of the poles have multiplicities, then one needs to construct the A_i matrices in their upper triangular form using the method described in [4, p. 284]. From here on we assume, without loss of generality, that none of the poles have any multiplicities and therefore all A_i matrices can be diagonalized. The state matrix X_d will be essential in computing the reduced Gröbner basis $G = \{g_1, \dots, g_k\}$. We now prove Theorem 3.1, which states that the polynomials of G do not depend on the initial state $x(\mathbf{0})$ and are invariant under the same similarity transformation of all A_i matrices. Theorem 3.1 hence implies that in order to find the polynomials G from X_d , one can construct X_d using almost any nonzero initial state $x(\mathbf{0})$ and with all A_i matrices in diagonal form.

Theorem 3.1 *Let $G = \{g_1, \dots, g_k\}$ be a set of state difference polynomials. Then these polynomials do not depend on the initial state $x(\mathbf{0})$ and are invariant under the similarity transformation $A_i = V^{-1} A_i V$ ($i = 1, \dots, m$), with V an arbitrary nonsingular matrix.*

Proof 1 *Suppose that the maximal degree of the polynomials in G is d . The fact that any $g \in G$ is a state difference polynomial can then be written as $X_d c = 0$, where c is a column vector containing all coefficients of g . Substitution of X_d with the right-hand side of (7) leads to*

$$g(A_1, \dots, A_m) x(\mathbf{0}) = 0. \quad (8)$$

Since the state difference polynomial has to satisfy (8) for any initial state $x(\mathbf{0})$, it follows that $g(A_1, \dots, A_m)$ has to be the zero matrix and g therefore does not depend on $x(\mathbf{0})$. Applying the similarity transformation V to all A_i matrices we have that

$$A_1^{t_1} \cdots A_m^{t_m} = V \bar{A}_1^{t_1} \cdots \bar{A}_m^{t_m} V^{-1}. \quad (9)$$

Using (9) we can write (8) as

$$V g(\bar{A}_1, \dots, \bar{A}_m) V^{-1} x(\mathbf{0}) = 0,$$

which implies that $g(\bar{A}_1, \dots, \bar{A}_m)$ also needs to be the zero matrix.

Observe from Theorem 3.1 that all A_i matrices are transformed under the same similarity transformation. This follows trivially from the fact that the A_1, \dots, A_m matrices commute and therefore have the same eigenvectors. The commutation requirement is easily understood

from the following simple 2D example. The state $x(1, 1)$ can be reached from $x(\mathbf{0})$ via two distinct paths. The first path is $A_1 A_2 x(\mathbf{0})$, while the alternative path is described by $A_2 A_1 x(\mathbf{0})$. Since the value of $x(1, 1)$ is independent from the path that was taken, it follows that $A_1 A_2 = A_2 A_1$. A generalization of the Cayley-Hamilton theorem to the regular overdetermined mD case follows from the fact that (8) is valid for any state difference polynomial $p \in \langle g_1, \dots, g_k \rangle$.

Corollary 3.1 *(regular mD Cayley-Hamilton theorem) For a set of generators $G = \{g_1, \dots, g_k\}$ of the ideal of state difference polynomials $\langle g_1, \dots, g_k \rangle$ we have that $p(A_1, \dots, A_m) = 0 \quad \forall p \in \langle g_1, \dots, g_k \rangle$.*

Since there are n standard monomials B , there will be n linearly independent states in X_d . This implies that $\text{rank } X_d = n$ for sufficiently large d and this adds an additional constraint on which $x(\mathbf{0})$ can be used to construct X_d .

Lemma 3.1 *Suppose none of the poles have multiplicities. Then $x(\mathbf{0})$ has to consist of n nonzero entries in order for $\text{rank } X_d = n$ to be valid for sufficiently large d .*

Proof 2 *The condition that none of the poles have multiplicities implies that all A_i can be chosen to be diagonal. Hence, each row j of X_d will be the sequence $\lambda_{j1}^{t_1} \cdots \lambda_{jm}^{t_m} x_j(\mathbf{0})$, where λ_{ji} stands for the i th entry of λ_j and $x_j(\mathbf{0})$ is the j th entry of $x(\mathbf{0})$. A zero-entry of $x(\mathbf{0})$ would therefore introduce a row of zeros in X_d such that the condition $\text{rank } X_d = n$ can never be attained for sufficiently large d .*

Lemma 3.1 is also valid when the poles have multiplicities. In this case, however, the condition on the entries of $x(\mathbf{0})$ is sufficient and not necessary.

3.1 Algorithm for regular mD systems

All ingredients are now in place to derive and present our algorithm that computes a reduced Gröbner basis of state difference polynomials for a given set of affine poles. From $X_d c = 0$ it is easily seen that the coefficient vectors of the desired Gröbner basis lie in the kernel of X_d . However, computing the kernel of X_d will in general not result in a Gröbner basis. First observe that we can also interpret X_d in the following way

$$X_d = \begin{pmatrix} x(\mathbf{0}) & z_1 x(\mathbf{0}) & z_2 x(\mathbf{0}) & \cdots & z_m^d x(\mathbf{0}) \end{pmatrix}, \quad (10)$$

which means that each state, or equivalently each column of X_d , can be associated with a monomial in the shift operators. We now need to find linear relations of the form (5) between the different states. In other words, the set of standard monomials B and leading monomials A need to be determined. The standard monomials are

the states of X_d that form a basis for the column space of X_d , $\text{col}(X_d)$, and correspond with monomials of lowest degree. Our algorithm will therefore test the columns of X_d , starting with $x(\mathbf{0})$, for linear independence with respect to all its previous linearly independent states. When a state $z^t x(\mathbf{0})$ is found to be linearly independent with respect to all its previous linearly independent states, then z^t is a standard monomial and added to B . Otherwise z^t is a leading monomial and hence defines a state difference polynomial. Indeed, suppose that during the execution of the algorithm for a 2D system we have that $B = \{1, z_1, z_2\}$ and that z_1^2 is found to be linearly dependent with respect to B . Interpreting B as a set of column indices of X_d , this can also be written using MATLAB notation as $\text{rank}(X_d(:, B)) = 3$. The first element of A is then $a = z_1^2$ and the first polynomial of the reduced Gröbner basis g_1 therefore has the form $g_1 = c_{20} z_1^2 + c_{01} z_2 + c_{10} z_1 + c_{00}$ with unknown coefficients c_{00}, \dots, c_{20} . These coefficients can be uniquely determined by solving the following linear system

$$\begin{pmatrix} X_d(:, B) & X_d(:, a) \end{pmatrix} \begin{pmatrix} c_{00} \\ c_{10} \\ c_{01} \\ c_{20} \end{pmatrix} = 0, \quad (11)$$

which is guaranteed to have a solution since (11) expresses the linear dependence of the column $X_d(:, a)$ with respect to the columns of $X_d(:, B)$. Observe that in this notation a is also interpreted as both the column index of X_d and the corresponding monomial. The matrix on the left-hand side of (11) can be written compactly using MATLAB notation as $X_d(:, [B, a])$. Since z_1^2 is now determined to be a leading monomial of a reduced Gröbner basis, it immediately follows that none of the states of X_d that are monomial multiples of z_1^2 need to be considered anymore. Checking whether a state is linearly independent and if it is not, solving the linear system (11) for the unknown coefficients is achieved by one SVD, which can be computed in a numerically backward stable manner [14]. When not all n standard monomials have been determined, a criterion is needed to decide whether the state a under investigation is linearly independent with respect to all other elements of B . This is done by inspection of the singular values of $X_d(:, [B, a])$, obtained from computing its SVD

$$X_d(:, [B, a]) = U \Sigma V^T$$

where U, V are orthogonal matrices and Σ is a diagonal matrix containing the singular values $\bar{\sigma} \geq \dots \geq \underline{\sigma}$. When the smallest singular value $\underline{\sigma}$ is smaller than a chosen tolerance τ , then a is determined to be linearly dependent. If we assume that the numerical values of the poles are known with infinite precision then the tolerance τ is set to $n \bar{\sigma} \epsilon$, where n is the order of the system,

$\bar{\sigma}$ is the largest singular value and ϵ is the machine precision, which is in double precision $\approx 10^{-16}$. If the poles are only known up to a certain accuracy, then the tolerance τ needs to take this into account. If the monomial a is found to be linearly dependent, then the right singular vector v corresponding with the smallest singular value is the vector of coefficients that solves (11). The smallest singular value $\underline{\sigma}$ then also serves as a measure of how well the retrieved polynomial vanishes on the given poles. Indeed, writing $X_d(:, [B, a])$ as the product of a column selection matrix S and X_d we have that

$$\|X_d(:, [B, a]) v\|_2 = \|X_d S v\|_2 = \|X_d c\|_2 = \underline{\sigma}.$$

The vector $c \triangleq S v$ is the desired coefficient vector of the reduced Gröbner basis polynomial after normalization such that the coefficient of the leading monomial is 1. The only remaining ingredient for the algorithm is a stop condition that tells us when d is large enough. This is easily derived, again using the defining property of Gröbner bases.

Lemma 3.2 *Let B be the set of standard monomials and A be the set of leading monomials determined from X_d . Furthermore, let \mathcal{T}_d^m be the set of all m -variate monomials of degree d . If the cardinality of B is n , the number of poles including multiplicities, and all elements of \mathcal{T}_{d+1}^m are divisible by elements of A , then d is sufficiently large.*

Proof 3 *Since the cardinality of B is n , no new standard monomials can appear at degrees $d+1$ or higher. If all elements of \mathcal{T}_{d+1}^m are divisible by elements of A , then all elements of \mathcal{T}_{d+k}^m for $k \geq 2$ are also divisible by elements of A . This implies that no other new linearly dependent monomials can be found anymore for any degree larger than $d+1$.*

The whole numerical SVD-based algorithm to determine a reduced Gröbner basis of state difference polynomials from a given set of affine poles is presented in Algorithm 1. Removing the monomial multiples of A limits the total number of SVDs to $n + n_A$, where n_A is the cardinality of A . Since the left singular vectors U do not need to be computed, the computational complexity of each SVD is approximately $4n(n+1)^2 + 8(n+1)^3$ flops [14, p. 254]. In fact, once the n standard monomials are known during the algorithm, no decision on the linear dependence of the remaining monomials needs to be made anymore. The only remaining task is therefore the computation of the right singular vector corresponding with the smallest singular value. We have not taken this optimization into account. The MATLAB/Octave implementation of Algorithm 1 in the PNLA package is `abma.m`.

Algorithm 1 *Reduced Gröbner basis G from poles \mathcal{Z}*

Input: set of poles \mathcal{Z} , monomial ordering $<$

Output: reduced Gröbner basis G

$A, G \leftarrow \emptyset, B \leftarrow 1, d \leftarrow 0$

```

construct  $X_d$ 
while Lemma 3.2 not satisfied do
   $d \leftarrow d + 1$ 
   $\mathcal{X} \leftarrow \mathcal{T}_d^m$ 
  remove all monomial multiples of  $A$  from  $\mathcal{X}$ 
  add states  $\mathcal{X}$  to  $X_d$ 
  for all monomials in  $\mathcal{X}$  do
     $a \leftarrow$  smallest monomial in  $\mathcal{X}$  according to  $<$ 
     $[U, \Sigma, V] \leftarrow \text{SVD}(X_d(:, [B, a]))$ 
     $\tau \leftarrow m \bar{\sigma} e$ 
    if  $\underline{\sigma} < \tau$  then
      append  $a$  to  $A$  and remove it from  $\mathcal{X}$ 
      normalize and append  $c \triangleq S \underline{v}$  to  $G$ 
    else
      append  $a$  to  $B$  and remove it from  $\mathcal{X}$ 
    end if
  end for
end while

```

4 Singular mD systems

In this section we will discuss the conceptual and algorithmic changes that are required to compute the reduced Gröbner basis state difference polynomials for singular systems. First, we define singular overdetermined mD systems.

Definition 4.1 *A singular (or descriptor) mD system is a system for which all poles \mathcal{Z} are points in the projective space \mathbb{P}^m and for which at least 1 pole lies at infinity.*

As was stated in Section 2, the introduction of poles at infinity require the use of homogeneous polynomials. This entails the introduction of an additional shift operator z_0 and the extension of \mathbf{t} with an extra variable t_0 . The autonomous overdetermined mD singular system is hence described by (1) over the $m+1$ -tuple \mathbf{t} with the additional equation $x(t_0 + 1, t_1, \dots, t_m) = A_0 x(\mathbf{t})$. Note that this is a whole new formulation of a descriptor system compared to the well-known descriptor system equations

$$E x(t_1, \dots, t_k + 1, \dots, t_m) = A_k x(t_1, \dots, t_m),$$

for $k = 1, \dots, m$ and some singular matrix E . Considering everything in terms of homogeneous polynomials has as a consequence that the state sequence matrix X_d for singular systems is given by

$$X_d = \begin{pmatrix} x(d, \dots, 0) & x(d-1, 1, \dots, 0) & \cdots & x(0, \dots, d) \end{pmatrix}, \\ = \begin{pmatrix} A_0^d x(\mathbf{0}) & A_0^{d-1} A_1 x(\mathbf{0}) & \cdots & A_m^d x(\mathbf{0}) \end{pmatrix},$$

and consists of all states $z^t x(\mathbf{0})$ with $\mathbf{t} \in \mathbb{N}_0^{m+1}$ and $|\mathbf{t}| = d$. This kind of state sequence where all states on the hyperplane $|\mathbf{t}| = d$ need to be considered separately for consecutive values of d is conceptually very

different from regular systems. Indeed, the resulting homogeneous state difference polynomials will allow us to compute the ‘next’ state on the hyperplane $|\mathbf{t}| = d$ from ‘previous’ states that are also on the same hyperplane. As a consequence, the set of linearly independent monomials B now also depend on d and will need to be recomputed for each iteration of the algorithm. Just like for regular systems, $\text{rank} X_d = n$ for sufficiently large d . Theorem 3.1 and its corollary, the mD generalization of the Cayley-Hamilton theorem, are equally valid for singular systems. Likewise, the condition of Lemma 3.1 on $x(\mathbf{0})$ to consist of all nonzero entries is also sufficient for singular systems to construct X_d with the A_i matrices diagonal or in upper triangular form. The stop condition of Lemma 3.2 does not apply to singular systems since the monomials of B are recomputed for each d . In order to formulate a new stop condition we need to introduce the concept of connected monomials.

Definition 4.2 ([1, p. 348]) *Let $B \subset \mathcal{T}_d^{m+1}$. Two monomials b, b' are connected in B if there is a sequence of monomials $b_0, b_1, \dots, b_r \in B$ with $b_0 = b$ and $b_r = b'$ such that for each $i = 1, \dots, r$ there exists $\alpha, \beta \in \{0, \dots, m\}$ satisfying $b_i = b_{i-1} \cdot x_\alpha / x_\beta$.*

Or in other words, two monomials in B are connected if one can pass from one b_i to the next by replacing one variable by another. We then call the connected component of a monomial $b \in B$ the set of all monomials in B that are connected with b . The stop condition for singular systems can now be formulated completely in terms of the connectivity of monomials in the set of linearly independent monomials.

Lemma 4.1 ([1, p. 349]) *Let B be the monomial set of linearly independent monomials determined from the homogeneous X_d . Then d is large enough if*

- (1) *the cardinality of B is n , the total number of projective poles including multiplicities and*
- (2) *for all $i = 0, \dots, m$, every monomial in the connected components of z_i^d in B is divisible by z_i .*

Another stop condition is formulated in terms of computing Hilbert Series and can be found in [20, p. 401]. The whole SVD-based projective Buchberger-Möller algorithm is easily derived from Algorithm 1. One needs to take into account that B needs to be recomputed for each value of d and use the stop condition of Lemma 4.1 instead. The projective Buchberger-Möller algorithm is implemented in the MATLAB/Octave PNLA package as pbma.m. The same remarks for Algorithm 1 on the tolerance, accuracy and computational complexity apply here as well. The total number of SVD’s is larger compared to the affine algorithm due to the fact that the standard monomials B are recomputed for each degree.

4.1 Separation of state variables

For singular systems, the poles can be split into 2 kinds: affine poles, characterized by having $z_0 = 1$, and poles at infinity with $z_0 = 0$. This distinction leads to a natural split of the state vector $x(\mathbf{t})$ into 2 parts. We will now demonstrate how this separation of the state variables comes about. We can partition the A_i matrices in the following way

$$A_0 = \begin{pmatrix} I & 0 \\ 0 & E_0 \end{pmatrix}, A_k = \begin{pmatrix} R_k & 0 \\ 0 & E_k \end{pmatrix} \quad (k = 1, \dots, m).$$

The distinction between affine poles and poles at infinity is clearly visible from the A_0 matrix. Indeed, the unit matrix I corresponds with the affine poles and the nilpotent E_0 with the poles at infinity. Note that the matrices E_1, \dots, E_m are not necessarily nilpotent. Let n_R and n_S denote the size of I and E_0 respectively. Then we have that $n_R + n_S = n$. This partitioning of the A_i matrices naturally induces a partitioning of the $n \times 1$ state vector $x(\mathbf{t})$ into a regular and singular part $x(\mathbf{t}) = \begin{pmatrix} x^R(\mathbf{t}) & x^S(\mathbf{t}) \end{pmatrix}^T$ with $x^R(\mathbf{t}) \in \mathbb{C}^{n_R}$ and $x^S(\mathbf{t}) \in \mathbb{C}^{n_S}$. Using (3) we can then write

$$\begin{pmatrix} x^R(\mathbf{t}) \\ x^S(\mathbf{t}) \end{pmatrix} = \begin{pmatrix} I^{t_0} R_1^{t_1} \dots R_m^{t_m} x^R(0) \\ E_0^{t_0} E_1^{t_1} \dots E_m^{t_m} x^S(0) \end{pmatrix}. \quad (12)$$

The separation of the state variables into a regular and singular part as in (12) shows that the regular part $x^R(\mathbf{t})$ indeed behaves as a regular mD system since $I^{t_0} x^R(0, t_1, \dots, t_m) = x^R(0, t_1, \dots, t_m)$. Also note that the nilpotence of E_0 has as a result that $x^S(\mathbf{t}) = 0$ for almost all values of \mathbf{t} . In the 1D case, this separation of state variables leads to 2 completely separate state sequences. Indeed, for $m = 1$ there is only 1 possible point at infinity, viz. $(0, 1)$, with a certain multiplicity. This results in the following A_i matrices

$$A_0 = \begin{pmatrix} I_{n_R} & 0 \\ 0 & E_0 \end{pmatrix}, A_1 = \begin{pmatrix} R_1 & 0 \\ 0 & I_{n_S} \end{pmatrix},$$

and state vector

$$x(t_0, t_1) = \begin{pmatrix} I_{n_R}^{t_0} R_1^{t_1} x^R(0, 0) \\ E_0^{t_0} I_{n_S}^{t_1} x^S(t_0, 0) \end{pmatrix} = \begin{pmatrix} x^R(0, t_1) \\ x^S(t_0, 0) \end{pmatrix}.$$

If we take for example $d = 3$, then the homogeneous state sequence is

$$X_3 = \begin{pmatrix} x(3, 0) & x(2, 1) & x(1, 2) & x(0, 3) \end{pmatrix}, \\ = \begin{pmatrix} x^R(0, 0) & x^R(0, 1) & x^R(0, 2) & x^R(0, 3) \\ x^S(3, 0) & x^S(2, 0) & x^S(1, 0) & x^S(0, 0) \end{pmatrix}.$$

Removing the spurious zero indices in the regular and singular part of the state vector we can write

$$X_3 = \begin{pmatrix} x^R(0) & x^R(1) & x^R(2) & x^R(3) \\ x^S(3) & x^S(2) & x^S(1) & x^S(0) \end{pmatrix},$$

which is in the literature commonly explained as a regular state sequence that runs forward in time and a singular state sequence that runs backward in time. This complete separation of the state sequence into a forward and backward evolving part is not possible in general for the mD case. The following lemma provides a condition on the E_i matrices such that this separation becomes possible.

Lemma 4.2 *The state sequence of an autonomous singular overdetermined mD system can be divided into a forward evolving regular part and backward evolving singular part if E_1, \dots, E_m are multiples of the unit matrix I .*

Proof 4 Suppose that $E_i = a_i I$ with $a_i \in \mathbb{C}$ for $i = 1, \dots, m$. Then

$$E_0^{t_0} E_1^{t_1} \dots E_m^{t_m} = a_1^{t_1} \dots a_m^{t_m} E_0^{t_0} \triangleq \tilde{E}_0^{t_0}$$

and therefore

$$x(t_0, \dots, t_m) = \begin{pmatrix} x^R(0, t_1, \dots, t_m) \\ x^S(t_0, 0, \dots, 0) \end{pmatrix}.$$

5 Numerical Experiments

In this section we will illustrate the application of our proposed algorithms to compute the state difference equations for both regular and singular mD systems. In addition, we illustrate how these systems can be realized from the difference equations via a simulation diagram. All experiments were run in MATLAB on a quad-core 3.30 GHz desktop with 16 GB RAM.

Experiment 1 *For a first simple example we want to realize a 2D system with the following set of affine poles*

$$\mathcal{Z} = \{(0.1268 - 0.7076i, -0.8759 + 0.2205i), \\ (0.1268 + 0.7076i, -0.8759 - 0.2205i), \\ (-4.5389, -2.5864), (0.0853, 2.9382)\}.$$

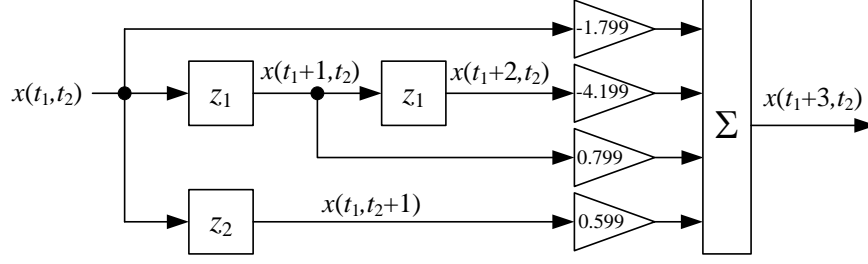


Fig. 1. Realization of (14) in Experiment 1 as a simulation diagram.

It can be immediately deduced that the resulting system will be internally unstable due to some of the entries of the poles being outside the unit disc. Running Algorithm 1 computes the following state difference equations

$$\begin{aligned} g_1 &= -0.333 + z_1 - 0.333 z_1^2 + z_1 z_2 = 0, \\ g_2 &= -2.666 - z_1 - 2 z_2 - 0.666 z_1^2 + z_2^2 = 0, \\ g_3 &= 1.799 - 0.799 z_1 - 0.599 z_2 + 4.199 z_1^2 + z_1^3 = 0, \end{aligned} \quad (13)$$

and finishes in 0.015 seconds. The leading monomials according to the graded xelicographic ordering are

$$A = \{LM(g_1) = z_1 z_2, LM(g_2) = z_2^2, LM(g_3) = z_1^3\}$$

and therefore we have that $B = \{1, z_1, z_2, z_1^2\}$, which implies that $x(0,0), x(1,0), x(0,1), x(2,0)$ are required to start up the state recursions. Observe that the number of B monomials equals the number of poles. In order to realize the autonomous system and describe all possible future states $x(\mathbf{t})$, we first need to rewrite each of the difference equations of (13) as $LM(z_1^{t_1} z_2^{t_2} g_i) = \sum_{j=1}^n \beta_j z_1^{t_1} z_2^{t_2} b_j$. For $g_3 = 0$, this results in the equation

$$\begin{aligned} z_1^{t_1+3} z_2^{t_2} &= -4.199 z_1^{t_1+2} z_2^{t_2} + 0.599 z_1^{t_1} z_2^{t_2+1} \\ &\quad + 0.799 z_1^{t_1} z_2^{t_2+1} - 1.799 z_1^{t_1} z_2^{t_2}, \end{aligned} \quad (14)$$

which is realized by means of delay, multiplication and addition elements in Figure 1. Similarly, g_1, g_2 can be rewritten and realized using delay, multiplication and addition elements.

Experiment 2 Next, we add 2 additional poles at infinity $(0,1,0), (0,0,1)$ to the poles \mathcal{Z} of Experiment 1 and run the projective version of Algorithm 1 to obtain the following 5 homogeneous state difference polynomials

$$\begin{aligned} g_1 &= -0.333 z_0^3 + z_0^2 z_1 - 0.333 z_0 z_1^2 + z_0 z_1 z_2, \\ g_2 &= -2.666 z_0^3 - z_0^2 z_1 - 2 z_0^2 z_2 - 0.666 z_0 z_1^2 + z_0 z_2^2, \\ g_3 &= 0.6 z_0^3 - 0.6 z_0^2 z_1 - 0.2 z_0^2 z_2 + 2.4 z_0 z_1^2 + z_1^2 z_2, \\ g_4 &= 0.533 z_0^3 - 1.12 z_0^2 z_1 - 0.4 z_0^2 z_2 + 1.133 z_0 z_1^2 + z_1 z_2^2, \\ g_5 &= 1.8 z_0^4 - 0.8 z_0^3 z_1 - 0.6 z_0^3 z_2 + 4.2 z_0^2 z_1^2 + z_0 z_1^3, \end{aligned} \quad (15)$$

in 0.102 seconds. The total degrees of g_1, g_2, g_3, g_4 and g_5 are 3 and 4 respectively. The autonomous mD

system described by the 5 homogeneous polynomials (15) is singular and has 6 poles. This implies always 6 states are required for every total degree d to compute all remaining states on the hyperplane $|\mathbf{t}| = d$. The number of states in X_0, X_1, X_2 are 1, 3, 6 respectively. It is therefore only possible to compute all states X_d for $d \geq 3$. For example, X_3 contains 10 states, 4 of which can be computed as the leading monomials of g_1, g_2, g_3, g_4 . The remaining 6 initial states $f X_3$ are $x(3,0,0), x(2,1,0), x(2,0,1), x(1,2,0), x(0,3,0), x(0,0,3)$. The conditions of Lemma 4.2 do not apply to this singular system and it is therefore not possible to interpret the state sequence into a forward evolving regular part and a backward evolving singular part.

6 Conclusions

The problem of computing the state difference polynomials of a discrete autonomous mD system from a given set of poles was solved by means of two algorithms. The classical Cayley-Hamilton theorem was in this way generalized to polynomial ideals. System theoretic interpretations were given to the notion of polynomial ideals, leading monomials and Gröbner bases. Furthermore, the inclusion of poles at infinity into the theory resulted in a new state space description of autonomous mD descriptor systems. The separation of the state vector into a regular and singular part was also discussed and conditions were given when this leads to a forward and backward evolving state sequence. The application of the two developed algorithms was illustrated in numerical experiments.

References

- [1] J. Abbott, A. Bigatti, M. Kreuzer, and L. Robbiano. Computing ideals of points. *J. Symb. Comput.*, 30(4):341 – 356, 2000.
- [2] S. Attasi. *Systèmes linéaires homogènes à deux indices*. Rapport de recherche / Institut de recherche d’informatique et d’automatique, Laboratoire de recherche en informatique et automatique. LABORIA, 1973.
- [3] K. Batselier, P. Dreesen, and B. De Moor. The Geometry of Multivariate Polynomial Division and Elimination. *SIAM J. Matrix Anal. A.*, 34(1):102–125, 2013.

- [4] K. Batselier, P. Dreesen, and B. De Moor. On the null spaces of the macaulay matrix. *Linear Algebra Appl.*, 460(0):259 – 289, 2014.
- [5] I. Bleylevens, R. Peeters, and B. Hanzon. An nD-systems approach to global polynomial optimization with an application to H2model order reduction. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 5107–5112, Dec 2005.
- [6] I. Bleylevens, R. Peeters, and B. Hanzon. Efficiency improvement in an nd systems approach to polynomial optimization. *J. Symb. Comput.*, 42(1-2):30–53, January 2007.
- [7] N.K. Bose. *Multidimensional Systems Theory and Applications*. Springer Netherlands, 2010.
- [8] B. Buchberger. Gröbner Bases and Systems Theory. *Multidimensional Systems and Signal Processing*, 12(3):223–251, July 2001.
- [9] D. A. Cox, J. B. Little, and D. O’Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics. Springer-Verlag, March 2005.
- [10] D. A. Cox, J. B. Little, and D. O’Shea. *Ideals, Varieties and Algorithms*. Springer-Verlag, third edition, 2007.
- [11] P. Dreesen. *Back to the Roots: Polynomial System Solving Using Linear Algebra*. PhD thesis, Faculty of Engineering, KU Leuven (Leuven, Belgium), 2013.
- [12] E. Fornasini and G. Marchesini. State-space realization theory of two-dimensional filters. *IEEE T. Automat. Contr.*, 21(4):484–492, Aug 1976.
- [13] E. Fornasini, P. Rocha, and S. Zampieri. State space realization of 2-d finite-dimensional behaviours. *SIAM J. Control Optim.*, 31(6):1502–1517, 1993.
- [14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, October 1996.
- [15] M. Hazewinkel and B. Hanzon. *Constructive algebra and systems theory*. Verhandelingen der Koninklijke Nederlandse Akademie van Wetenschappen, Afd. Natuurkunde. Royal Netherlands Academy of Arts and Sciences, 2006.
- [16] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse. Approximate Computation of Zero-dimensional Polynomial Ideals. *J. Symb. Comput.*, 44(11):1566–1591, November 2009.
- [17] B.L. Ho and R.E. Kalman. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 14(1-12):545–548, 1966.
- [18] T. Kaczorek. The singular general model of 2d systems and its solution. *IEEE T. Automat. Contr.*, 33(11):1060–1061, Nov 1988.
- [19] T. Kaczorek. Generalization of Cayley-Hamilton theorem for n-D polynomial matrices. *IEEE T. Automat. Contr.*, 50(5):671–674, May 2005.
- [20] M. Kreuzer and L. Robbiano. *Computational Commutative Algebra 2*. Springer, 2005.
- [21] J.E. Kurek. The general state-space model for a two-dimensional linear digital system. *IEEE T. Automat. Contr.*, 30(6):600–602, Jun 1985.
- [22] M.S. Livšic, N. Kravitsky, A.S. Markus, and V. Vinnikov. *Theory of Commuting Nonselfadjoint Operators*. Mathematics and Its Applications. Springer, 1995.
- [23] David G. Luenberger. Time-invariant Descriptor Systems. *Automatica*, 14(5):473–480, September 1978.
- [24] P. Misra, P. Van Dooren, and V. Syrmos. Pole-zero representation of descriptor systems. *Automatica*, 31(6):907 – 912, 1995.
- [25] H.M. Möller and B. Buchberger. The construction of multivariate polynomials with preassigned zeros. In Jacques Calmet, editor, *Computer Algebra*, volume 144 of *Lecture Notes in Computer Science*, pages 24–31. Springer Berlin Heidelberg, 1982.
- [26] R. Roesser. A discrete state-space model for linear image processing. *IEEE T. Automat. Contr.*, 20(1):1 – 10, February 1975.
- [27] J. Rosenthal and D.S. Gilliam. *Mathematical Systems Theory in Biology, Communications, Computation and Finance*. IMA volumes in mathematics and its applications. Springer, 2003.
- [28] L. Shaul and V. Vinnikov. State feedback for overdetermined 2d systems: pole placement for bundle maps over an algebraic curve. In *Multidimensional (nD) Systems, 2009. nDS 2009. International Workshop on*, pages 1–2, June 2009.
- [29] N.J. Theodorou. M-dimensional Cayley-Hamilton theorem. *IEEE T. Automat. Contr.*, 34(5):563–565, May 1989.
- [30] E. Zerz. The discrete multidimensional mpum. *Multidim. Syst. Sign. P.*, 19(3-4):307–321, 2008.